

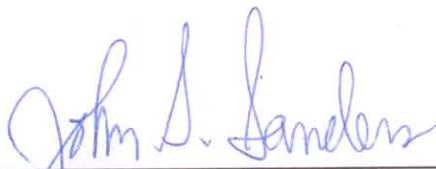
STANDARD OPERATING PROCEDURE
LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

KEY WORDS

Database, Well inventory database, Surface water database, pesticide chemistry database, data entry

APPROVALS

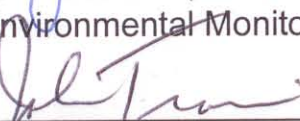
APPROVED BY:


John Sanders, Ph.D.
Environmental Monitoring Branch Management

DATE:

8/24/05

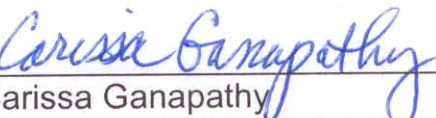
APPROVED BY:


John Troiano, Ph.D.
Environmental Monitoring Branch Senior Scientist

DATE:

8/22/05


APPROVED BY:


Carissa Ganapathy
Environmental Monitoring Branch Quality Assurance Officer

DATE:

8/22/05

PREPARED BY:


Jeff Schuette
Environmental Research Scientist

DATE:

8/17/05

Environmental Monitoring Branch organization and personnel, such as management, senior scientist, quality assurance officer, project leader, etc., are defined and discussed in SOP ADMN002.

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

1.0 INTRODUCTION

1.1 Purpose

The Environmental Monitoring (EM) Branch receives pesticide monitoring and environmental fate data from DPR monitoring studies as well as from other State and local agencies. Typically, EM receives data in the form of a textfile or a Microsoft Excel file, which was created in the Windows environment. EM must then transfer these data to existing Oracle tables (collectively forming a database), which reside in a UNIX environment. If the textfile consists of only a few rows of data, individual rows of data can be manually extracted from the textfile and entered into the Oracle tables using the SQL, INSERT command. For larger data sets, this is a tedious task and there are distinct disadvantages to using the INSERT command. For one, the INSERT command does not allow multiple rows of data to be entered into a table. Additionally, the INSERT command does not produce a log file. The log file can be used as a record to document that the data was loaded and it also provides detailed information about data that did not load correctly. This standard operating procedure (SOP) serves as a guide for loading large quantities of data from textfiles into existing UNIX based Oracle tables using the Windows environment or the UNIX environment. This method produces a log file that effectively tracks all data loaded into the Oracle tables.

1.2 Definitions

- 1.2.1 **Delimited Textfile**- A textfile in which each of the data elements within a row are separated by a character or space. This method of formatting files is recognized by popular spreadsheets and database applications facilitating importing of data between applications.
- 1.2.2 **Control File** - This is a file containing programming instructions for transferring data from a textfile into an existing Oracle table using commands from SQL.
- 1.2.3 **Database (Relational)** - A method of structuring data as collections of tables that are logically associated to each other by shared attributes. Any data element in a database is searchable by

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

knowing the name of the table, the attribute (column) name, and the shared value (primary key). The EM Branch manages three Oracle databases, the Well Inventory, the Surface Water, and the Pesticide Chemistry database.

- 1.2.4 **Instance-** A technical term used to refer to a database in a way that is inclusive of all of its technical components i.e., the disk files that hold the tables, the part of the computer memory allocated for the database and the Oracle software that is necessary to manage the database. EM currently uses the server, Scalos, which has three instances, EMON, PUR1, PUR2.
- 1.2.5 **SQL Loader-** The specific program that uses the control file to physically transfer the data from a textfile into the existing ORACLE table .
- 1.2.6 **SQL-** Structured Query Language. A standard language for searching and modifying relational databases.
- 1.2.7 **Syntax-** Refers to the command language structure that executes specific functions. Italicized words in this SOP represents information supplied by the user and that will change with the users unique criteria. Examples are a pathname, filename, username, or password. Non-italized text are those words required by the command.
- 1.2.8 **Telnet-** Telnet is a utility program and protocol that allows one to connect to another computer on a network. After providing a username and password to login to the remote computer, one can enter commands that will be executed as if entered directly from the remote computer's console.
- 1.2.9 **Textfile-** A file that consists of text characters with limited formatting information. Data elements within the row of a textfile are often delineated by a space or a comma. Textfiles are also known as an ASCII files. This file can be opened in any word processor. Textfiles can be incompatible across operating systems because different operating systems use different markers to indicate the end of a line.

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

2.0 PROCEDURES

2.1 General outline for procedures

- 2.1.1 A textfile containing the data is created. Each row of text contains the data for one record. For well sampling, each record represents a unique chemical analysis. The methodology to create a delimited textfile is explained in section 2.2. Skip this section if the textfile has already been created.
- 2.1.2 A control file is created using SQL syntax to transfer data from a textfile into an existing Oracle table. Control files that load delimited textfiles are explained in section 2.3.1 and for textfiles with no delimiters in section 2.3.3.
- 2.1.3 The SQL Loader program invokes the control file to transfer the data into the existing Oracle tables. The SQL loader program can be run from either Windows as explained in section 2.4 or the UNIX environment as explained in section 2.5.

2.2 Creating textfiles

To create textfiles, data are gathered from printed materials such as reports, chain-of-custody forms and data sheets. The data is then entered into a spreadsheet using a program such as Microsoft EXCEL. The following procedure is used for EXCEL spreadsheets:

- 2.2.1 In the first row of the EXCEL spreadsheet, consecutive columns are given the same name as the fieldnames in the Oracle table into which the data will be stored.
- 2.2.2 All spreadsheet cells should be formatted as text or number. By convention, text formatting is usually chosen for all columns.
- 2.2.3 Oracle uses the following date format, DD-MMM-YYYY i.e. 02-JAN-2005. Dates entered into the spreadsheet should be in this format.
- 2.2.4 The last named column in the spreadsheet should be a column that will always contain data i.e. sampling date. There are, however, two other approaches to handling null data entered into the last column:

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

- 2.2.4.1 In the last column of the textfile, enter the word 'NULL' into cells that contain no data (Do not use the parenthesis).
- 2.2.4.2 Alternative, the statement "Trailing Nullcols" can be included into the control file used to load the data. This will handle the issue with the last column not always containing data.
- 2.2.5 Upon completion of the spreadsheet, save the file as an Excel spreadsheet denoted as a .xls filetype. This file will be used as a back up.
- 2.2.6 Lastly, the file will be saved again but this time as a delimited textfile. First, delete the first row in the spreadsheet, which contains the field names. Then save the Excel spreadsheet as either a tab delimited or comma delimited textfile. Section 2.2.6.1 identifies the selections in Excel, from the 'save type as' drop down menu that will create a tab delimited file and section 2.2.6.2 are selections that will create a comma delimited file.
 - 2.2.6.1 Selections for tab delimited textfiles
 - Text (Tab delimited) (*.txt)
 - Text (MS-DOS) (*.txt)
 - 2.2.6.2 Selections for comma delimited textfiles
 - CSV (comma delimited) (*.csv)
 - CSV (MS-DOS) (*.csv)

2.3 Creating Control Files

The Oracle loader program uses a control file to transfer the data from textfiles into existing Oracle tables. For textfiles that were created without using procedure 2.2 above, verify that the textfile is somehow delimited. Open the textfile in any word processor and verify that each element in the file is separated by a delimiter like a character or space. If the file is not delimited and some or all of the elements have no separation, then proceed to procedure 2.3.3, which creates a control file for textfiles without delimiters. Otherwise, if the textfile is a delimited file, proceed as follows in section 2.3.1, which creates a control file for a delimited textfile.

2.3.1 Control file for delimited textfile

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

2.3.1.1 Open a text editor like Notepad or Wordpad, and type the following **bolded** six lines of text:

2.3.1.1.1 **Load data**

2.3.1.1.2 **infile 'filepath\yourtextfile.csv'**

2.3.1.1.3 **into table *Oracletablename*** [Note: if the table that you are entering data into is not empty, either empty the table or add the word 'append' before the word 'into' Warning! Do not empty the permanent Oracle tables.]

2.3.1.1.4 **fields terminated by ',' optionally enclosed by '''**
[Note: this statement identifies the delimiter in the first set of quotes. If your file is not a comma delimited file then replace the comma with the delimiter for your file.]

2.3.1.1.5 **trailing nullcols** [Note: this statement is optional (see 2.2.4.2)]

2.3.1.1.6 **(*fieldname1, fieldname2, etc..*)** [Note: starting from left to right, the field names should exactly match the column names that you created in your original Excel spreadsheet or textfile.]

2.3.1.2 Save by clicking save as and in the 'save as type:' drop down menu select 'all files', then, in the 'filename' box type in a name for the control file adding the extension 'ctl' (i.e. *filename.ctl*).

2.3.2 Example of a control file for a comma delimited textfile:

```
----- top of file -----  
LOAD DATA  
  INFILE 'h:\groundwater2005.csv'  
  INTO TABLE tempwidata  
  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ''''  
  TRAILING NULLCOLS  
  (<well, agency, conc, rpt_yr>  
  |
```

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

2.3.3 Control file for textfiles that are not delimited

If the textfile is not a delimited file, open the textfile in a text editor like Notepad or Wordpad. Identify which elements in the textfile that correspond to the field names in your existing Oracle tables. Counting from left to right, identify the position number of the first character and last character of each element.

2.3.3.1 Open a text editor like Notepad or Wordpad, and type the following three **bolded** lines of text:

2.3.3.1.1 **Load data**

2.3.3.1.2 **infile 'filepath\yourtextfile.csv'**

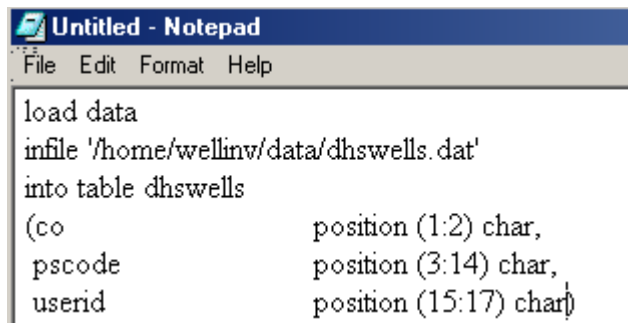
2.3.3.1.3 **into table *Oracletablename*** [Note: if the table that you are entering data into is not empty either empty the table or add the word 'append' before the word 'into'. Warning! Do not empty the permanent Oracle tables.]

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

- 2.3.3.2 On the next line, in parenthesis, type in the *field name* from the Oracle table followed by the word, 'position'. Then, in parenthesis, type in the first and last position number of the element that corresponds to that field name. Follow this by typing in the word, 'char,' (see example 2.3.4).
- 2.3.3.3 Save by clicking save as and in the 'save as type:' drop down menu select 'all files', then, in the 'filename' box type in a name for the control file adding the extension 'ctl' (i.e. *filename.ctl*).

2.3.4 Example of a control file for any textfile.



```
load data
infile '/home/wellinv/data/dhswells.dat'
into table dhswells
(co                position (1:2) char,
pscode            position (3:14) char,
userid            position (15:17) char)
```

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

2.4 Loading data into an Oracle Table using Microsoft Windows

NOTE: it is best to load your data into temporary Oracle tables and later, transfer the data to the permanent tables. In the Windows environment, the SQL loader program (SQLldr) must have been installed when the SQLPLUS utility was installed on the desktop.

2.4.1 Open a command line window by clicking start>run then typing 'command' and hit the 'enter' key.

2.4.2 Run the loader program using your control file

2.4.2.1 At the command line prompt, type 'sqlldr
username/password@instance
control='filepath\controlfilename.ctl', then hit the 'enter' key.
Note: the username and password are associated with the owner of the existing Oracle tables and are most likely not your personal username and password. For example, the ground water database is owned by WELLINV. The username/password would be wellinv/password@emon.

2.4.3 Once the data loads, a log file is created and stored in the same location as the control file. The log file is given the same name as the control file but with the extension 'log'. It provides details about how the file was loaded and if any errors occurred. If there were errors, a second file, the bad file, is created and stored where the textfile was stored. This file is given the name of your textfile but with the extension 'bad'. This file will include all the rows that did not load due to the error.

2.5 Loading data using the UNIX environment

2.5.1 NOTE: it is best to load your data into temporary Oracle tables and later, transfer the data to the permanent tables. Log on to the UNIX server using Telnet.

2.5.2 Save or transfer your textfile (csv file) and the control file (ctl file) to your UNIX directory.

2.5.3 Convert your textfile from a DOS format to a UNIX format. At the UNIX command line, type in 'DOS2UNIX *textfilename.csv*
newtextfilename.dat' then hit the 'enter' key. DOS2UNIX is a

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

conversion tool used to remove a carriage return that exists in textfiles created in the Windows environment. By entering the name of your textfile then entering a new textfile name with the extension 'dat', you are left with 2 files, the original DOS 'csv' file and the new UNIX 'dat' file. Use the 'dat' file to load the data into the Oracle tables. This ensures a smoother transition of data from your textfiles into the Oracle tables.

2.5.4 Run the loader program using your control file as follows:

2.5.4.1 At the command line prompt, type 'sqlldr
username/password control='filepath\controlfilename.ctl',
then hit the 'enter' key. Note: the username and password
are associated with the owner of the existing Oracle tables
and are most likely not your personal username and
password. For example, the ground water database is
owned by WELLINV. The username/password would be
wellinv/password.

2.5.5 Once the data loads, a log file is created and stored in the same location as the control file. The log file is given the same name as the control file but with the extension 'log'. It provides details about how the file was loaded and if any errors occurred. If there were errors, a second file, the bad file, is created and stored where the textfile was stored. This file is given the name of your textfile but with the extension 'bad'. This file will include all the rows that did not load due to the error.

2.6 When errors occur

2.6.1 Open the log file. The log file is very good at identifying why the data did not load correctly. Once you identify the error(s) you have two options. The first is to fix the error(s) in your original Excel spreadsheet re-save the spreadsheet to a textfile then reload the new textfile. The second option is to open the bad file that contains all the data that did not load and fix the error in the bad file. Then, load the bad file into the Oracle tables by changing the filename (2.3.1.1.2) in the control file to reflect the location of the bad file.

STANDARD OPERATING PROCEDURE

LOADING LARGE DATASETS INTO ORACLE 8i TABLES FROM UNIX AND WINDOWS

2.6.2 Common Errors

- 2.6.2.1 There were cells in the last column of data that were blank and the control file did not include the statement 'Trailing Nullcols'.
- 2.6.2.2 The field names identified in the control file were not in the same order as the original Excel spreadsheet.
- 2.6.2.3 There are cells in the Excel spreadsheet outside of your dataset that contain data or a space, which you may not have noticed.
- 2.6.2.4 When loading data from the UNIX environment, the textfile was not converted to UNIX format.
- 2.6.2.5 For control files created for any textfile, the positioning of the element did not match the the Oracle field name.